# Data Layer Documentation

## Table of Contents

* [ArrayStore](#arraystore)
* [CustomStore](#customstore)
* [DataSource](#datasource)
* [LocalStore](#localstore)
* [Query](#query)

## ArrayStore

ArrayStore is a client-side in-memory store that holds an array of data objects.

### Important Options

* **data**: An array of data objects to be stored.
* **key**: The key property or properties by which data objects are identified.
* **onInserted**: A function that is executed after a data object is added.
* **onRemoved**: A function that is executed after a data object is removed.
* **onUpdated**: A function that is executed after a data object is updated.

### Methods

* **insert(values)**: Adds a new data object.
* **remove(key)**: Removes a data object.
* **update(key, values)**: Updates a data object.
* **load(options)**: Loads data objects.
* **byKey(key)**: Gets a data object by key.

### Events

* **onLoaded**: Fires after data is loaded.
* **onLoading**: Fires before data is loaded.
* **onModified**: Fires after a data object is modified.

## CustomStore

CustomStore allows you to implement custom data access logic.

### Important Options

* **load**: A function that is executed to load data.
* **byKey**: A function that is executed to get a data object by key.
* **insert**: A function that is executed to add a data object.
* **update**: A function that is executed to update a data object.
* **remove**: A function that is executed to remove a data object.

### Methods

* **load(options)**: Loads data objects using the custom load logic.
* **byKey(key)**: Gets a data object by key using the custom logic.
* **insert(values)**: Adds a new data object using the custom logic.
* **update(key, values)**: Updates a data object using the custom logic.
* **remove(key)**: Removes a data object using the custom logic.

### Events

* **onLoaded**: Fires after data is loaded using the custom logic.
* **onLoading**: Fires before data is loaded using the custom logic.
* **onInserted**: Fires after a data object is added using the custom logic.
* **onRemoved**: Fires after a data object is removed using the custom logic.
* **onUpdated**: Fires after a data object is updated using the custom logic.

## DataSource

DataSource is an object that provides data to UI components.

### Important Options

* **store**: Specifies the data store from which to load data.
* **filter**: Specifies a filter to be applied to data.
* **sort**: Specifies sorting options.
* **group**: Specifies grouping options.
* **select**: Specifies data fields to select.
* **expand**: Specifies whether to load all data or only the first page.

### Methods

* **load()**: Loads data from the store.
* **reload()**: Reloads data from the store.
* **dispose()**: Disposes of the DataSource instance.
* **isLoaded()**: Checks whether data is loaded.
* **isLoading()**: Checks whether data is being loaded.

### Events

* **changed**: Fires after data is changed.
* **loadError**: Fires when an error occurs during data loading.
* **loadingChanged**: Fires when the loading state changes.

## LocalStore

LocalStore is a client-side store that uses the local storage or session storage to persist data.

### Important Options

* **name**: The name of the data store.
* **immediate**: Specifies whether to save changes immediately.
* **key**: The key property or properties by which data objects are identified.

### Methods

* **load()**: Loads data from the local storage or session storage.
* **insert(values)**: Adds a new data object.
* **update(key, values)**: Updates a data object.
* **remove(key)**: Removes a data object.
* **clear()**: Clears all data from the local storage or session storage.

### Events

* **onLoaded**: Fires after data is loaded from the local storage or session storage.
* **onLoading**: Fires before data is loaded from the local storage or session storage.
* **onInserted**: Fires after a data object is added.
* **onRemoved**: Fires after a data object is removed.
* **onUpdated**: Fires after a data object is updated.

## Query

Query is a utility that allows you to perform complex queries on an array or a data store.

### Important Options

* **array**: The array of data objects to query.

### Methods

* **filter(predicate)**: Filters data objects based on a predicate.
* **sortBy(field)**: Sorts data objects by a field.
* **groupBy(field)**: Groups data objects by a field.
* **select(selector)**: Projects data objects into a new form.
* **aggregate(aggregator)**: Aggregates data objects.
* **toArray()**: Converts the query result to an array.

### Events

* **none**: Query does not have specific events.